Amendments to the Claims

Listing of Claims:

1. (Previously presented) A method for controlling a hardware circuit with a processor, the processor used for executing a program code to control the hardware circuit, the program code comprising:

a plurality of lower-level subroutines, wherein after the processor executes various lower-level subroutines, the hardware circuit will be controlled to execute various corresponding operations, and each the lower-level subroutine will record operation results, which come from the hardware circuit executing corresponding operations, in an error code; wherein each operation result corresponds to a recovery operation;

a plurality of higher-level subroutines, each higher-level subroutine used for calling at least a lower-level subroutine to control the hardware circuit to execute operations corresponding to the lower-level subroutine called by the higher-level subroutine when the processor executes the higher-level subroutine;

a plurality of recovery subroutines, each recovery subroutine corresponding to a recovery operation, wherein the hardware circuit is controlled to execute various corresponding recovery operations after the processor executes various recovery subroutines; and

an error-handling subroutine for calling the recovery subroutines according to the error code;

the method comprising:

after the processor executes the higher-level subroutine, executing the error-handling subroutine to allow the processor to control the hardware circuit to execute recovery operations according to the operation results corresponding to the lower-level subroutine called by the higher-level subroutine.

2. (Previously presented) The method of claim 1, wherein when the processor executes the error-handling subroutine after the higher-level subroutine is executed,

the processor will not execute the recovery operations corresponding to the lower-level subroutine called by the higher-level subroutine until the higher-level subroutine is finished.

3. (Original) The method of claim 1, wherein the higher-level subroutines won't call each other so that a next higher-level subroutine will not be executed until the processor finishes executing a previous higher-level subroutine.

4. (Original) The method of claim 1, wherein the hardware circuit is a servo module of an optical storage drive, the servo module comprising:
a motor for driving an optical disk to rotate; and
a pick-up head for generating a laser incident on the optical disk.

5. (Original) The method of claim 1, wherein the hardware circuit is an interface module of an optical storage drive.

6. (Previously presented) The method of claim 1, wherein the error code is a global variable of the program code; the operation results corresponding to the lower-level subroutines will be recorded in the same error code.

7. (Previously presented) The method of claim 1, wherein the program code further comprises a plurality of next-level subroutines; when the processor executes various next-level subroutines, the hardware circuit is controlled to execute corresponding operations; each next-level subroutine will record operation results corresponding to the hardware circuit in a second error code; each lower-level subroutine is used for calling at least a next-level subroutine so that the processor executes the next-level subroutine called by the lower-level subroutine to control the hardware circuit to execute corresponding operations when executing the lower-level subroutine.

8. (Previously presented) The method of claim 7, wherein next-level subroutines called by each lower-level subroutine record corresponding operation results in the same second error code.

9. (Original) The method of claim 7, wherein the second error code is a column of the error code.

10. (Original) The method of claim 7, wherein the next-level subroutines record corresponding operation results in the same second error code.

11. (Canceled)

12. (Original) The method of claim 1, wherein the lower-level subroutines won't call each other so that a next lower-level subroutine will not be executed until the processor finishes executing a previous lower-level subroutine.

13. (Original) The method of claim 1, wherein the lower-level subroutines won't call the higher-level subroutines.

14. (Previously presented) An electronic device, comprising:
a hardware circuit for achieving operations of the electronic device;
a processor for executing a program code to control the hardware circuit;
a storage device for storing the program code; wherein the program code comprises:
a plurality of lower-level subroutines, wherein after the processor executes various lower-level subroutines, the hardware circuit will be controlled to execute various corresponding operations, and each lower-level subroutine will record operation results, which come from the hardware circuit executing corresponding operations, in an error code; wherein each operation result corresponds to a recovery operation;

a plurality of higher-level subroutines, each higher-level subroutine used for calling at least a lower-level subroutine to control the hardware circuit to execute operations corresponding to the lower-level subroutine called by the higher-level subroutine when the processor executes the higher-level subroutine;

a plurality of recovery subroutines, each recovery subroutine corresponding to a recovery operation, wherein the hardware circuit is controlled to execute various corresponding recovery operations after the processor executes various recovery subroutines; and

an error-handling subroutine for calling the recovery subroutines according to the error code;

wherein after executing the higher-level subroutine, the processor executes the error-handling subroutine to allow the processor to control the hardware circuit to execute recovery operations according to the operation results corresponding to the lower-level subroutine called by the higher-level subroutine.

15. (Previously presented) The electronic device of claim 14, wherein when the processor executes the error-handling subroutine after the higher-level subroutine is executed, the processor will not execute the recovery operations corresponding to the lower-level subroutine called by the higher-level subroutine until the higher-level subroutine is finished.

16. (Original) The electronic device of claim 14, wherein the higher-level subroutines won't call each other so that a next higher-level subroutine will not be executed until the processor finishes executing a previous higher-level subroutine.

17. (Original) The electronic device of claim 14 being an optical storage drive, the hardware circuit comprising a servo module, which comprising:

a motor for driving an optical disk to rotate; and

a pick-up head for generating a laser incident on the optical disk.

18. (Original) The electronic device of claim 14 being an optical storage drive, the hardware circuit being an interface module of the optical storage drive.

19. (Previously presented) The electronic device of claim 14, wherein the error code is a global variable of the program code; the operation results corresponding to the lower-level subroutines will be recorded in the same error code.

20. (Previously presented) The electronic device of claim 14, wherein the program code further comprises a plurality of next-level subroutines; when the processor executes various next-level subroutines, the hardware circuit is controlled to execute corresponding operations; each next-level subroutine will record operation results corresponding to the hardware circuit in a second error code; each lower-level subroutine is used for calling at least a next-level subroutine so that the processor executes the next-level subroutine called by the lower-level subroutine to control the hardware circuit to execute corresponding operations when executing the lower-level subroutine.

21. (Previously presented) The electronic device of claim 20, wherein next-level subroutines called by each lower-level subroutine record corresponding operation results in the same second error code.

22. (Original) The electronic device of claim 20, wherein the second error code is a column of the error code.

23. (Original) The electronic device of claim 20, wherein the next-level subroutines record corresponding operation results in the same second error code.

24. (Canceled)

25. (Original) The electronic device of claim 14, wherein the lower-level subroutines won't call each other so that a next lower-level subroutine will not be executed until the processor finishes executing a previous lower-level subroutine.

26. (Original) The electronic device of claim 14, wherein the lower-level subroutines won't call the higher-level subroutines.

27. (New) The method of claim 1, wherein the error-handling subroutine unifies and manages recovery operations of all subroutines included in the program code, except the error-handling subroutine.

28. (New) The electronic device of claim 14, wherein the error-handling subroutine unifies and manages recovery operations of all subroutines included in the program code, except the error-handling subroutine.